General approach:

: you apply your favorite base selection procedures to subsamples of the data then you aggregate. Do with amny splits of the data , subsamples then youe eventaulyl select the variables that keep getting chosen over and over agin in these subssampels.

where they prove a finite sample upper bound on the expected umber of falsely selected samples you end up with through this procedure. You can use this upper bound to determine the threshold of the proportion of times a variable has to occur in the asubsample for you to choose it eventually

The limitation of L1-based sparse models is that faced with a group of very correlated features, they will select only one. To mitigate this problem, it is possible to use randomization techniques, reestimating the sparse model many times perturbing the design matrix or sub-sampling data and counting how many times a given regressor is selected.

[Stability selection](http://stat.ethz.ch/~nicolai/stability.pdf) is a relatively novel method for feature selection, based on subsampling in combination with selection algorithms (which could be regression, SVMs or other similar method). The high level idea is to apply a feature selection algorithm on different subsets of data and with different subsets of features. After repeating the process a number of times, the selection results can be aggregated, for example by checking how many times a feature ended up being selected as important when it was in an inspected feature subset. We can expect strong features to have scores close to 100%, since they are always selected when possible. Weaker, but still relevant features will also have non-zero scores, since they would be selected when stronger features are not present in the currently selected subset, while irrelevant features would have scores (close to) zero, since they would never be among selected features.

introduces a new method called stability selection whose goal is to provide an algorithm for performing model selection in a structure learning problem while controlling the number of false discoveries.

The Stability Selection theory, recently proposed by [[21](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#R21)] is a general approach to address problems related to variable selection or discrete structure estimation (as graphs or clusters). The properties of this approach are particularly beneficial for applications involving high dimensional data, specially in cases where the number of variables or covariates *p* largely exceeds the number of examples *n* (i.e. the *p* >> *n* case).

In the stability selection framework, data are perturbed several times (for example by iterative sub-sampling the examples). For each perturbation, a method that produces sparse coefficients is applied to a sub-sample of the data. After a large number of iterations, all features that were selected in a large fraction of the perturbations are chosen. Finally a cutoff threshold (0 < *thr* < 1) is applied in order to select the most stable features.

Stability selection is a general framework to combine variable selection methods such as penalized regression models with subsampling strategies. Variable selection probabilities are estimated by applying variable selection methods to subsamples of the data, drawn without replacement, and estimating the proportion of subsamples where the variable was included in the fitted model. These selection probabilities are used to define a set of stable variables. [Meinshausen and Bühlmann (2010)](http://bioinformatics.oxfordjournals.org/content/early/2015/04/28/bioinformatics.btv197.full#ref-17) provide a theoretical framework for controlling Type I error rates of falsely assigning variables to the set of stable variables. Here we suggest to apply the subsampling scheme of stability selection to the lassoestimator involved in the RSPCA algorithm to estimate selection probabilities which are then used to identify the truly relevant variables contributing to a PC. As the lasso selects true variables with high probability the corresponding selection probabilities estimated with stability selection are expected to dominate those of irrelevant variables. Applying a classical forward model selection to the features ranked by these selection probabilities, sparse loadings vectors that are parameter estimation consistent as well as model selection consistent can be identified.

If yo do l1 penalization you end up with some components of your estimator parameter vector being 0 so you can do this automatic simultaneous variable selection and parameter estimation. Stability selection is not in itself a variable selection method but it’s one that youo can bolt onto any existing variable seleciotn technique to improve its performance. Can be used in conjucuton with any base variable selction procedure and any underlying data generating mechanism. The idea is simple: you apply your favorite base selection procedures to subsamples of the data then you aggregate. Do with amny splits of the data , subsamples then youe eventaulyl select the variables that keep getting chosen over and over agin in these subssampels.

In paper introduce idea of feature of method where they prove a finite sample upper bound on the expected umber of falsely selected samples you end up with through this procedure. You can use this upper bound to determine the threshold of the proportion of times a variable has to occur in the asubsample for you to choose it eventually

The model im assuming is very straighotofrwatd, have some data. I.i.d . some are signal varibales some are noise vairbales. Not worry too muc by what I mean with signal and noise in this talk. Imagine each random vector can be artitioned as a corvariate and reposne pair with a covariate I p dimensional space and a real valued response. Imagine you ave som log likelihood of the form. Signal variable are component are non zero noise are zero coefficients. I

The limitation of L1-based sparse models is that faced with a group of very correlated features, they will select only one. To mitigate this problem, it is possible to use randomization techniques, reestimating the sparse model many times perturbing the design matrix or sub-sampling data and counting how many times a given regressor is selected.

[Stability selection](http://stat.ethz.ch/~nicolai/stability.pdf) is a relatively novel method for feature selection, based on subsampling in combination with selection algorithms (which could be regression, SVMs or other similar method). The high level idea is to apply a feature selection algorithm on different subsets of data and with different subsets of features. After repeating the process a number of times, the selection results can be aggregated, for example by checking how many times a feature ended up being selected as important when it was in an inspected feature subset. We can expect strong features to have scores close to 100%, since they are always selected when possible. Weaker, but still relevant features will also have non-zero scores, since they would be selected when stronger features are not present in the currently selected subset, while irrelevant features would have scores (close to) zero, since they would never be among selected features.

Randomized Logistic Regression We use randomized logistic regression to assess the importance of features. Our model uses 27 features to model stopout. In order to best fit a training set, a logistic regression model optimizes weights for each feature. To assess the importance of the features randomized logistic regression repeatedly models a perturbed data set (sub-sample) with regularization, and works as follows:

Step 1: Sample without replacement 75% of the training data each time (the variables are normalized ahead of training).

Step 2: Train a logistic regression model on the subsampled data, with randomized regularization coefficient for each variable.

The randomized coefficient βj is sampled from uniform distribution [λ, λ α ], where α ∈ (0, 1] and λ is the regularization coefficient usually used in standard regularized regression approaches. This randomization places different selection pressure for different variables.

Step 3: For every covariate evaluate b j s = µ(wj , th) where µ is a unit step function and wj is the

coefficients for covariate i and th is the threshold we set to deem the feature important.

This is set at 0.25.

Thus this results in a binary vector, that represents the selection of the covariate.

This binary vector is (lag × |features|) long where 1 at a location j implies feature i = j mod 27 was present in this model.

Step 4: Repeat Steps 1, 2 and 3 a total of 200 times.

Step 5: Estimate the importance of the covariate j by calculating the selection probabilities P s b j s 200 .

introduces a new method called stability selection whose goal is to provide an algorithm for performing model selection in a structure learning problem while controlling the number of false discoveries.

It works in the high-dimensional data setting (p n), which is currently a very active area of research in statistics and machine learning. 2. It provides control on the family-wise error rate in the finite sample setting, which is more practical than an asymptotic guarantee.

Let’s assume that we have a generic structure estimation algorithm that takes a dataset Z = Z1, . . . , Zn and a regularization parameter λ and returns a selection set Sˆλ . We can think of this algorithm as a black box. The stability selection algorithm then runs as follows:

1. Define a candidate set of regularization parameters Λ and a subsample number N.

2. For each value of λ ∈ Λ, do:

(a) Start with the full dataset Z(full) = Z1, . . . , Zn

(b) For each i in 1, . . . , N, do:

i. Subsample from Z(full) without replacement to generate a smaller dataset of size bn/2c, given by Z(i) .

ii. Run the selection algorithm on dataset Z(i) with parameter λ to obtain a selection set Sˆλ (i) .

(c) Given the selection sets from each subsample, calculate the empirical selection probability for each model component:

Πˆ λ k = P{k ∈ Sˆλ } = 1 N X N i=1 I{k ∈ Sˆλ i }.

The selection probability for component k is its probability of being selected by the algorithm.

3. Given the selection probabilities for each component and for each value of λ, construct the stable set according to the following definition:

Sˆstable = {k : max λ∈Λ Πˆ λ k ≥ πthr}

where πthr is a predefined threshold.

Note that this procedure doesn’t simply find the best value of λ ∈ Λ and then use it in the algorithm, but actually identifies a set of “stable” variables that are selected with high probability. The authors state that the empirical results vary little for threshold values in the range (0.6, 0.9), and are also not sensitive to choices of Λ.

The Stability Selection theory, recently proposed by [[21](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#R21)] is a general approach to address problems related to variable selection or discrete structure estimation (as graphs or clusters). The properties of this approach are particularly beneficial for applications involving high dimensional data, specially in cases where the number of variables or covariates *p* largely exceeds the number of examples *n* (i.e. the *p* >> *n* case).

In the stability selection framework, data are perturbed several times (for example by iterative sub-sampling the examples). For each perturbation, a method that produces sparse coefficients is applied to a sub-sample of the data. After a large number of iterations, all features that were selected in a large fraction of the perturbations are chosen. Finally a cutoff threshold (0 < *thr* < 1) is applied in order to select the most stable features.

According to the stability selection theory, for every set *K* ⊆ 1, ⋯ , *p*, the probability of *K* being in the selected set Sˆλ(I) is defined as

ΠˆλK=P∗(K⊆Sˆλ(I))

(2)

where *I* is a random subsample of 1, ⋯ , *n* of size n2 drawn without replacement. According to [[21](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#R21)], the probability *P*\* in the definition 2 regards both the random sub-sampling and other sources of randomness.

It is important to emphasize that according to stability selection theory, any regression method which produces sparse results can be used to select the features, as one is interested in the frequency of selections and not in the sparsity inherent to specific methods.

In [[21](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#R21)], the authors used the LASSO to demonstrate the properties of the stability selection framework in an application to select relevant features in a vitamin gene expression data set. The data set consisted of 115 examples and 4088 features. The authors permuted 4082 features and applied stability selection to find the remaining six relevant features.

The original formulation of the stability selection theory is based on sub-sampling of examples (as in bootstrapping procedures). However, the authors also proposed a modified version of the original framework, which they called *Randomised LASSO*. In this approach, instead of penalizing the absolute value *βk* of every component with a penalty term proportional to *λ* (as in [equation 1](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#FD1)), the Randomised LASSO changes the penalty *λ* to a randomly chosen value in a predefined range, according to the following equation:

βλˆ,W=argminβ∈Rp‖Y−Xβ‖22+λ∑k=1p∣βk∣Wk

(3)

The re-weighting is not based on any previous estimate, but is simply chosen randomly. According to [[21](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4576737/#R21)], applying this random re-weighting several times and looking for variables that are chosen often will turn out to be a very powerful procedure. They showed the superiority of Randomized LASSO in relation to the original stability selection formulation in the vitamin data set. Using the Randomized LASSO the six non-permuted features were selected and much less permuted features were included in the selected set (i.e. the number of false positive selections was lower than in the original formulation).

Stability selection is a general framework to combine variable selection methods such as penalized regression models with subsampling strategies. Variable selection probabilities are estimated by applying variable selection methods to subsamples of the data, drawn without replacement, and estimating the proportion of subsamples where the variable was included in the fitted model. These selection probabilities are used to define a set of stable variables. [Meinshausen and Bühlmann (2010)](http://bioinformatics.oxfordjournals.org/content/early/2015/04/28/bioinformatics.btv197.full#ref-17) provide a theoretical framework for controlling Type I error rates of falsely assigning variables to the set of stable variables. Here we suggest to apply the subsampling scheme of stability selection to the lassoestimator involved in the RSPCA algorithm to estimate selection probabilities which are then used to identify the truly relevant variables contributing to a PC. As the lasso selects true variables with high probability the corresponding selection probabilities estimated with stability selection are expected to dominate those of irrelevant variables. Applying a classical forward model selection to the features ranked by these selection probabilities, sparse loadings vectors that are parameter estimation consistent as well as model selection consistent can be identified.

In each experiment 200 logistic regression models are formed, thus adding up to a total of approximately 72,000 logistic regression models. For each experiment, randomized logistic regression resulted in a vector of covariates selection probabilities. Each of these probabilities ranged from 0 to 1. 7 Randomized logistic regression analysis gave us fascinating covariate selection probability vectors for all 91 experiments and all cohorts. For each experiment the randomized logistic regression gives us these selection probability vectors for all the covariates which are learner features for different weeks. In order to gain a more quantitative grasp of which features matter for different prediction problem, we aggregate these probabilities.